

FAILURE ANALYZING METHOD, AND ANALYSIS OF FAILURE SUPPORTING DEVICE AND PROGRAM

Publication number: JP2003194890

Publication date: 2003-07-09

Inventor: SUMITOMO HIROSHI

Applicant: NEC ELECTRONICS CORP

Classification:

- international: G01R31/28; G06F11/22; G01R31/28; G06F11/22;
(IPC1-7): G01R31/28; G06F11/22

- European:

Application number: JP20010395361 20011226

Priority number(s): JP20010395361 20011226

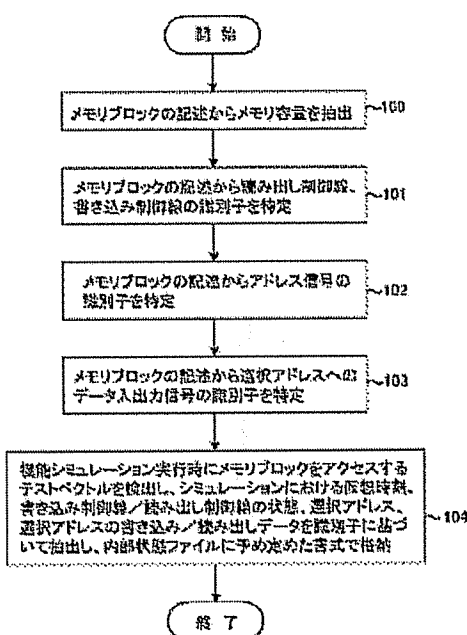
Report a data error here

Abstract of JP2003194890

PROBLEM TO BE SOLVED: To follow up a failure propagation route in a case when a memory block having only the functional description and not having the information on a gate description level exists in a failure signal propagation route.

SOLUTION: Identifiers of write control, read control and an address and data input and output wire or signal are specified in steps 101-103 with respect to the memory block having functional description, a simulation time, a write control wire/read control wire state corresponding to the time, a selected address, and the data to be written/read out with respect to the selected address are extracted on the basis of these identifiers, and outputted with a format of binary digit display when the simulation is performed in a step 104. In the failure analysis, the failure propagation route is followed up by comparing the output with the created memory block internal state output.

COPYRIGHT: (C)2003,JPO



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-194890

(P2003-194890A)

(43) 公開日 平成15年7月9日 (2003.7.9)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 1 R 31/28		C 0 6 F 11/22	3 1 0 F 2 G 1 3 2
G 0 6 F 11/22	3 1 0		3 3 0 B 5 B 0 4 8
	3 3 0		3 5 0 F
	3 5 0		3 6 0 C
	3 6 0	C 0 1 R 31/28	B
審査請求 未請求 請求項の数6 O L (全 14 頁) 最終頁に続く			

(21) 出願番号 特願2001-395361(P2001-395361)

(22) 出願日 平成13年12月26日 (2001. 12. 26)

(71) 出願人 302062931

NECエレクトロニクス株式会社

神奈川県川崎市中原区下沼部1753番地

(72) 発明者 住友 洋志

東京都港区芝五丁目7番1号 日本電気株式会社社内

(74) 代理人 100109313

弁理士 机 昌彦 (外2名)

Fターム(参考) 2G132 AA08 AB02 AC10 AC14 AL09

5B048 AA19 AA20 CC02 DD08 DD15

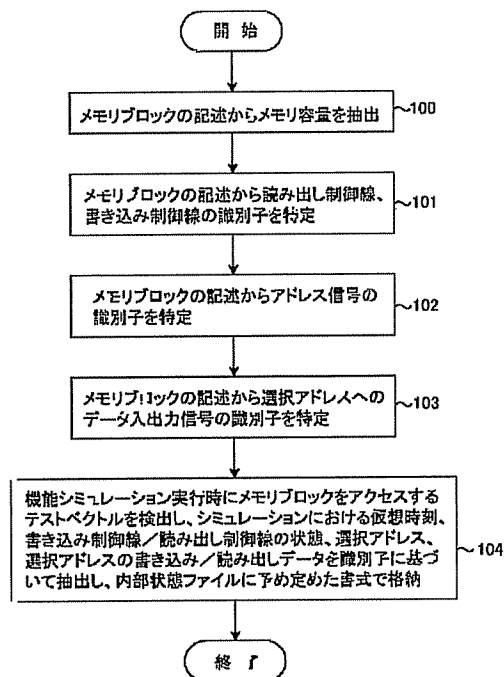
FF02

(54) 【発明の名称】 故障解析方法、故障解析支援装置および故障解析支援プログラム

(57) 【要約】

【課題】機能記述のみでゲート記述レベルの情報を持たないメモリブロックが故障信号の伝播経路上に存在する場合に、故障伝播経路の追跡を実用的な労力および時間で実行する。

【解決手段】機能記述されたメモリブロックについてステップ101~103で、書き込み制御、読み出し制御、アドレスおよびデータ入出力の配線もしくは信号の識別子を特定し、ステップ104で、シミュレーション時にこれら識別子に基づいてシミュレーション時刻とその時刻に対応する書き込み制御線/読み出し制御線の状態、選択アドレス、選択アドレスに書き込み/読み出しされるデータのそれぞれを抽出して、2進数表示のフォーマットで出力する。故障解析では、作成されたメモリブロック内部状態出力と比較して故障伝搬経路を追跡する。



【特許請求の範囲】

【請求項1】 機能記述されたメモリブロックを有するLSIの故障解析方法において、メモリブロックの機能記述から書き込み制御、読み出し制御、アドレスおよびデータ入出力の配線もしくは信号の識別子を特定する手順と、

LSIの機能シミュレーション実行時に前記メモリブロックをアクセスするテストベクトルを検出する毎に前記テストベクトルに対応する時刻における書き込み制御信号および読み出し制御信号の状態と、選択アドレスの2進数表現および該選択アドレスの入出力データの2進数表現とを前記識別子に基づいて抽出し所定の形式の内部状態リストを作成する手順と、

故障LSIの故障伝搬経路に前記メモリブロックが存在する場合に書き込み制御信号、読み出し制御信号、選択アドレスおよび該選択アドレスの入出力データの実測値と前記内部状態リスト中の対応する期待値との一致照合を行って故障伝搬経路を追跡し故障箇所を特定する手順と、を有することを特徴とする故障解析方法。

【請求項2】 機能記述されたメモリブロックを有するLSIの故障解析支援装置において、メモリブロックの機能記述を入力し、書き込み制御、読み出し制御、アドレスおよびデータ入出力の配線もしくは信号の識別子を特定する識別子特定手段と、

LSIの機能シミュレーション実行時に前記メモリブロックをアクセスするテストベクトルを検出する毎に前記テストベクトルに対応する時刻における書き込み制御信号および読み出し制御信号の状態と、選択アドレスの2進数表現および該選択アドレスの入出力データの2進数表現とを前記識別子に基づいて抽出し所定の形式の内部状態情報として出力する内部状態抽出手段と、を備えることを特徴とする故障解析支援装置。

【請求項3】 機能記述されたメモリブロックを有するLSIの故障解析を支援するコンピュータプログラムにおいて、メモリブロックの機能記述を入力し、書き込み制御、読み出し制御、アドレスおよびデータ入出力の配線もしくは信号の識別子を特定する手順と、

LSIの機能シミュレーション実行時に前記メモリブロックをアクセスするテストベクトルを検出する毎に前記テストベクトルに対応する時刻における書き込み制御信号および読み出し制御信号の状態と、選択アドレスの2進数表現および該選択アドレスの入出力データの2進数表現とを前記識別子に基づいて抽出し所定の形式の内部状態情報として出力する手順と、をコンピュータに実行させることを特徴とする故障解析支援プログラム。

【請求項4】 機能記述されたメモリブロックを有するLSIの故障解析方法において、機能テストで故障を出力するまでに発生したメモリブロックへアクセスとアクセス対象のアドレスとを全て抽出

する手順と、

レイアウトデータにおけるメモリブロックのデータ入出力線からトレースを開始しワード線数以上の分岐を有する分岐配線に到達した場合にトレースを停止して前記データ入出力線から前記分岐配線までの経路のレイアウトデータを抽出する手順と、

全てのアクセスに対して、それぞれのアクセスでのアドレス入力をもとにアドレス入力線から選択ワード線への経路を特定し前記選択ワード線から前記分岐配線に到達するまでトレースして経路に存在するレイアウトデータを抽出する手順と、

前記アドレス入力線から前記分岐配線までのレイアウトデータと前記データ入出力線から前記分岐配線までの経路のレイアウトデータとからアクセスにより選択されたメモリセルに係る部分回路のトランジスタレベルのネットリストを作成し、それぞれのアクセスに対応する部分回路を合成して置換用トランジスタレベルネットリストを作成する手順と、

メモリブロックの機能記述を前記置換用トランジスタレベルネットリストに置き換えたLSIの設計情報記述を用いて故障の解析を行う手順と、を有することを特徴とする故障解析方法。

【請求項5】 機能記述されたメモリブロックを有するLSIの故障解析支援装置において、機能テストで故障を出力するまでに発生したメモリブロックへアクセスとアクセス対象のアドレスとを全て抽出するアクセスアドレス保持手段と、

レイアウトデータにおけるメモリブロックのデータ入出力線からトレースを開始しワード線数以上の分岐を有する分岐配線に到達した場合にトレースを停止して前記データ入出力線から前記分岐配線までの経路のレイアウトデータを抽出したのちに、全てのアクセスに対して、それぞれのアクセスでのアドレス入力をもとにアドレス入力線から選択ワード線への経路を特定し前記選択ワード線から前記分岐配線に到達するまでトレースして経路に存在するレイアウトデータを抽出する経路データ抽出手段と、

前記アドレス入力線から前記分岐配線までのレイアウトデータと前記データ入出力線から前記分岐配線までの経路のレイアウトデータとからアクセスにより選択されたメモリセルに係る部分回路のトランジスタレベルのネットリストを作成し、それぞれのアクセスに対応する部分回路を合成して置換用トランジスタレベルネットリストを作成するネットリスト作成手段と、を備えることを特徴とする故障解析支援装置。

【請求項6】 機能記述されたメモリブロックを有するLSIの故障解析を支援するコンピュータプログラムにおいて、機能テストで故障を出力するまでに発生したメモリブロックへアクセスとアクセス対象のアドレスとを全て抽出

する手順と、

レイアウトデータにおけるメモリブロックのデータ入出力線からトレースを開始しワード線数以上の分岐を有する分岐配線に到達した場合にトレースを停止して前記データ入出力線から前記分岐配線までの経路のレイアウトデータを抽出する手順と、

全てのアクセスに対して、それぞれのアクセスでのアドレス入力をもとにアドレス入力線から選択ワード線への経路を特定し前記選択ワード線から前記分岐配線に到達するまでトレースして経路に存在するレイアウトデータを抽出する手順と、

前記アドレス入力線から前記分岐配線までのレイアウトデータと前記データ入出力線から前記分岐配線までの経路のレイアウトデータとからアクセスにより選択されたメモリセルに関係する部分回路のトランジスタレベルのネットリストを作成し、それぞれのアクセスに対応する部分回路を合成して置換用トランジスタレベルネットリストを作成する手順と、をコンピュータに実行させることを特徴とする故障解析支援プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はLSIの故障解析に関し、特に機能記述されたメモリブロックを有するLSIの故障解析に有効な故障解析方法、故障解析支援装置および故障解析支援プログラムに関する。

【0002】

【従来の技術】設計上はレイアウト検証まで問題なく終了し製造が開始されたLSIでも、製造時の不具合によって故障が作り込まれ不良品が発生する場合がある。このような場合に、故障箇所と原因を特定するために物理解析が実施される。物理解析において故障伝播経路がメモリに到達した場合には、機能検証シミュレーションでアクセスしたメモリアドレス、メモリセルの保持する値、書き込み制御線、読み出し制御線のそれぞれの期待値、または、良品のメモリセルのメモリアドレス、メモリセルの保持する値、書き込み制御線、読み出し制御線の実測値と、不良品のメモリアドレス、メモリセルの保持する値、書き込み制御線、読み出し制御線の実測値とを比較し、故障信号をメモリセルに伝播した原因がそのメモリセルを指定したアドレス、指定アドレスのメモリセルに保持されていたデータ、データの書き込み、またはデータの読み出しのいずれであるかを検証する。

【0003】一般に、三層以上の多層配線構造を持つ製品の配線層および拡散層で生じた全ての故障を非破壊で解析することは難しい。特に混載メモリのようにレイアウト配置が予め決定されているハードマクロでは、電位観測が可能な最上層に配線が存在することは稀である。そのため、FIB(Focused Ion Beam)等を用いて測定対象の配線表面にまで到達する微細な穴を形成し、下層配線の電気信号の情報を取得することを繰り返して故障箇

所を追跡する。

【0004】しかしながら、故障信号の伝播経路によってFIBによる微細穴を形成して故障箇所を追跡するには多大な解析時間を要する。この物理的な故障信号の追跡に要する時間を短縮するために、設計情報とシミュレーションによる期待値とを利用し、テスト時に検出された故障出力を追跡の起点としてソフトウェア上で故障伝播経路を追跡し故障箇所を特定する方法が知られている。一方法として、故障が存在する可能性のある組合せ回路を抽出しながら、故障伝播経路の追跡を行うことにより設計情報とテスト結果からフリップフロップの機能故障以外の故障箇所を特定する逆論理展開法がある。例えば、特開2001-21618号公報にはバックトラック法を応用した逆論理展開方法が開示されており、逆論理展開方法によって求めた入力信号の特定論理状態、およびその過程で求めた組合せ論理回路内部の論理状態と、予め正常回路を論理シミュレーションすることで求めた各ノードの論理状態(以下、期待値と称す)を比較することで故障伝播経路を抽出する。また別の方法として、あらかじめ故障シミュレーションを実行して故障辞書を作成しておき、診断時には故障辞書に記録された故障データと比較することで故障箇所を特定する方法がある。SCAN設計に特化したものであるが、故障辞書法により故障要因を検索する機能を持つソフトウェアが市販されている。

【0005】トップダウン設計方式における機能記述設計では、仕様設計をもとにLSI内部を機能ブロックに分割し、機能ブロック内のレジスタレジスタ間に設ける組合せ回路を決定し、クロックタイミング毎の回路動作を決定する。次の設計段階であるゲートレベル記述設計で機能ブロックのゲートレベル記述が完成するまでの間、シミュレーション等によるチップ全体の設計検証に用いられる。しかし、エンベディッドメモリなどのあらかじめレイアウトまで決定しているハードマクロブロックでは状況が異なる。ハードマクロブロックは内部回路のテスト手段を別途保有しており、機能が予め検証されている。このため、シミュレーションによるチップ全体の機能検証においては、どの設計段階においても機能記述された機能ブロックが使用される。

【0006】このようにハードマクロ化されて機能記述レベルの情報のみを持ちゲート記述レベルの情報を持たないメモリブロックがLSI内に存在し、且つメモリをアクセスするテストの実行時にエラー信号が出力された場合には、エラー信号がメモリブロックを経由して出力されたか否かを検証するために、まず、アクセスしたアドレスのメモリセルが保持するデータを参照する必要がある。メモリセルの保持するデータを参照する方法の一例として、メモリの設計仕様に対応したシミュレーションモデルを構築し、メモリデバッグ用検証ツールを用いてシミュレーション実行中にメモリ内容の観測、エラー

検出を行なうことにより、故障信号がメモリを経由したか否かを検証することができる。故障信号がメモリを経由していない場合には先に挙げた特開2001-21618号公報に記載の故障箇所特定方法等を適用することによって故障箇所の特定が可能になる。

【0007】

【発明が解決しようとする課題】電子ビームテストを用いた故障解析においては、良品と不良品との対応するレイアウト座標の電位像を比較することによって、パタン認識で一致しない信号線を故障信号伝播配線として容易に検出できる。しかし、期待値と実測値とを比較する場合には事情が異なる。シミュレーションにおける期待値はシミュレーション実行時にシミュレーション内で経過する時間として定義された仮想時間を指定して特定の配線が保持する値を取得したものである。期待値は仮想時間の経過にもなって変化した各ネット名とその保持する値を参照する書式で記述されるか、または、ネット毎に仮想時間経過に応じて変化した値を参照する書式で記述されるので、シミュレーションの期待値と不良品の実測値とをパタン認識的な簡便さで検証することができない。例えば、メモリブロックの設計情報の記述は図3に示すようにHDL (Hardware Description Language) で機能記述されているため、比較する信号の対応をとることが容易ではない。シミュレーションの期待値と不良品の実測値との比較ではなく、良品の実測値と不良品の実測値とを比較して故障箇所追跡する場合には、実測のために微細穴を形成する加工時間が2倍となり、故障解析に要する時間が大幅に増大する。

【0008】また、設計情報の記述において機能記述とゲートレベル記述とが混在しているLSIでは、機能記述された機能ブロックはゲートレベルの接続が未定義なので、先に挙げた故障辞書法で必要となる故障データの埋め込みに関して、メモリセルの保持する値が1または0の何れかに固定されるスタック故障以外の故障の埋め込みはできない。

【0009】また、各メモリセルの機能は図5に示すようにフリップフロップに置換した論理ゲートでも表現できるが、この場合にメモリのアドレスの指定を間違えるということは、フリップフロップの制御線から故障が伝播することである。先に挙げた逆論理展開法を基にした故障診断法では、フリップフロップの制御線が故障した場合には故障伝播経路の追跡ができないので、故障箇所を検出できない。すなわち、これまでの故障診断用ソフトウェアでは、特定のメモリセルがスタック故障である場合を除けば、機能記述されたメモリブロック内に故障信号が伝播する場合の故障箇所の検出は困難である。

【0010】また、メモリデバッグ用検証ツールを用いてシミュレーション実行中にメモリ内容を観測して検証するCADツールが市販されているが、このCADツールでは任意のシミュレーション時刻でアドレスが保持す

るセルの期待値を取得できるものの、誤信号がメモリのアドレス線から伝播したのかビット線から伝播したのか判別する手段を持たないため、メモリの故障箇所を特定するツールとして使うことは難しい。

【0011】レイアウトデータからネットリストを抽出するソフトウェアが知られており、これを用いてメモリブロックのネットリストを抽出し、メモリブロックの機能記述をトランジスタレベルのネットリストメモリに置き換えてシミュレーションすることも可能ではある。しかしながら、例えば、機能記述されたメモリブロックのメモリが100kbの比較的小容量のスタティックRAMの場合でも、トランジスタ数に換算すると百万トランジスタ程度となるので、メモリブロックの全トランジスタをネットリスト抽出する処理に時間がかかり、また、トランジスタレベルのシミュレーションにも膨大な時間がかかってしまうために実用的ではない。

【0012】このように、公知の故障解析方法では、ハードマクロ化されて機能記述レベルの情報のみを持ちゲート記述レベルの情報を持たないメモリブロックをアクセスするテストの実行時にエラー信号が出力された場合に、実用的なコストで実用的な時間内に故障箇所を特定することが困難であった。

【0013】本発明の目的は、機能記述のみでゲート記述レベルの情報を持たないメモリブロックが故障信号の伝播経路上に存在する場合においても、メモリブロック内部のアクセスされたアドレスのメモリセルにまで立ち入った故障伝播経路の追跡を実用的な労力および時間で実行できる故障解析支援装置および故障解析方法を提供することにある。

【0014】

【課題を解決するための手段】本発明の第1の発明の故障解析方法は、機能記述されたメモリブロックを有するLSIの故障解析方法において、メモリブロックの機能記述から書き込み制御、読み出し制御、アドレスおよびデータ入出力の配線もしくは信号の識別子を特定する手順と、LSIの機能シミュレーション実行時に前記メモリブロックをアクセスするテストベクトルを検出する毎に前記テストベクトルに対応する時刻における書き込み制御信号および読み出し制御信号の状態と、選択アドレスの2進数表現および該選択アドレスの入出力データの2進数表現とを前記識別子に基づいて抽出し所定の形式の内部状態リストを作成する手順と、故障LSIの故障伝播経路に前記メモリブロックが存在する場合に書き込み制御信号、読み出し制御信号、選択アドレスおよび該選択アドレスの入出力データの実測値と前記内部状態リスト中の対応する期待値との一致照合を行って故障伝播経路を追跡し故障箇所を特定する手順と、を有している。

【0015】第2の発明の故障解析支援装置は、機能記述されたメモリブロックを有するLSIの故障解析支援

装置において、メモリブロックの機能記述を入力し、書き込み制御、読み出し制御、アドレスおよびデータ入出力の配線もしくは信号の識別子を特定する識別子特定手段と、LSIの機能シミュレーション実行時に前記メモリブロックをアクセスするテストベクトルを検出する毎に前記テストベクトルに対応する時刻における書き込み制御信号および読み出し制御信号の状態と、選択アドレスの2進数表現および該選択アドレスの入出力データの2進数表現とを前記識別子に基づいて抽出し所定の形式の内部状態情報として出力する内部状態抽出手段と、を備えている。

【0016】第3の発明の故障解析支援プログラムは、機能記述されたメモリブロックを有するLSIの故障解析を支援するコンピュータプログラムにおいて、メモリブロックの機能記述を入力し、書き込み制御、読み出し制御、アドレスおよびデータ入出力の配線もしくは信号の識別子を特定する手順と、LSIの機能シミュレーション実行時に前記メモリブロックをアクセスするテストベクトルを検出する毎に前記テストベクトルに対応する時刻における書き込み制御信号および読み出し制御信号の状態と、選択アドレスの2進数表現および該選択アドレスの入出力データの2進数表現とを前記識別子に基づいて抽出し所定の形式の内部状態情報として出力する手順と、をコンピュータに実行させることを特徴としている。

【0017】第4の発明の故障解析方法は、機能記述されたメモリブロックを有するLSIの故障解析方法において、機能テストで故障を出力するまでに発生したメモリブロックへアクセスとアクセス対象のアドレスとを全て抽出する手順と、レイアウトデータにおけるメモリブロックのデータ入出力線からトレースを開始しワード線数以上の分岐を有する分岐配線に到達した場合にトレースを停止して前記データ入出力線から前記分岐配線までの経路のレイアウトデータを抽出する手順と、全てのアクセスに対して、それぞれのアクセスでのアドレス入力をもとにアドレス入力線から選択ワード線への経路を特定し前記選択ワード線から前記分岐配線に到達するまでトレースして経路に存在するレイアウトデータを抽出する手順と、前記アドレス入力線から前記分岐配線までのレイアウトデータと前記データ入出力線から前記分岐配線までの経路のレイアウトデータとからアクセスにより選択されたメモリセルに関する部分回路のトランジスタレベルのネットリストを作成し、それぞれのアクセスに対応する部分回路を合成して置換用トランジスタレベルネットリストを作成する手順と、メモリブロックの機能記述を前記置換用トランジスタレベルネットリストに置き換えたLSIの設計情報記述を用いて故障の解析を行う手順と、を有している。

【0018】第5の発明の故障解析支援装置は、機能記述されたメモリブロックを有するLSIの故障解析支援

装置において、機能テストで故障を出力するまでに発生したメモリブロックへアクセスとアクセス対象のアドレスとを全て抽出するアクセスアドレス保持手段と、レイアウトデータにおけるメモリブロックのデータ入出力線からトレースを開始しワード線数以上の分岐を有する分岐配線に到達した場合にトレースを停止して前記データ入出力線から前記分岐配線までの経路のレイアウトデータを抽出したのちに、全てのアクセスに対して、それぞれのアクセスでのアドレス入力をもとにアドレス入力線から選択ワード線への経路を特定し前記選択ワード線から前記分岐配線に到達するまでトレースして経路に存在するレイアウトデータを抽出する経路データ抽出手段と、前記アドレス入力線から前記分岐配線までのレイアウトデータと前記データ入出力線から前記分岐配線までの経路のレイアウトデータとからアクセスにより選択されたメモリセルに関する部分回路のトランジスタレベルのネットリストを作成し、それぞれのアクセスに対応する部分回路を合成して置換用トランジスタレベルネットリストを作成するネットリスト作成手段と、を備えている。

【0019】第6の発明の故障解析支援プログラムは、機能記述されたメモリブロックを有するLSIの故障解析を支援するコンピュータプログラムにおいて、機能テストで故障を出力するまでに発生したメモリブロックへアクセスとアクセス対象のアドレスとを全て抽出する手順と、レイアウトデータにおけるメモリブロックのデータ入出力線からトレースを開始しワード線数以上の分岐を有する分岐配線に到達した場合にトレースを停止して前記データ入出力線から前記分岐配線までの経路のレイアウトデータを抽出する手順と、全てのアクセスに対して、それぞれのアクセスでのアドレス入力をもとにアドレス入力線から選択ワード線への経路を特定し前記選択ワード線から前記分岐配線に到達するまでトレースして経路に存在するレイアウトデータを抽出する手順と、前記アドレス入力線から前記分岐配線までのレイアウトデータと前記データ入出力線から前記分岐配線までの経路のレイアウトデータとからアクセスにより選択されたメモリセルに関する部分回路のトランジスタレベルのネットリストを作成し、それぞれのアクセスに対応する部分回路を合成して置換用トランジスタレベルネットリストを作成する手順と、をコンピュータに実行させることを特徴としている。

【0020】

【発明の実施の形態】本発明について図面を参照して詳細に説明する。本発明の第1実施形態では、機能記述されたメモリブロックについて、書き込み制御、読み出し制御、アドレスおよびデータ入出力の配線もしくは信号の識別子を特定し、シミュレーション時にこれら識別子に基づいてシミュレーション時刻とその時刻に対応する書き込み制御線/読み出し制御線の状態、選択アドレ

ス、選択アドレスに書き込み／読み出しされるデータのそれぞれを抽出し、メモリブロック内部状態としてパターン認識的に容易に理解できる所定のフォーマットで出力する。また、信号線の電位実測による故障伝搬経路追跡では、故障伝搬経路にメモリブロックがあるときには、作成されたメモリブロック内部状態出力と比較して故障追跡する。

【0021】図1は、第1実施形態におけるメモリブロックの内部状態ファイルの作成フロー図である。図2は、内部状態ファイルを作成する故障解析支援装置を含むシステムのブロック図である。

【0022】図1において、まず、ステップ100で、機能記述されたメモリブロックからメモリ容量（アドレス空間）を抽出する。次のステップ101で、メモリブロックの読み出し制御線および書き込み制御線の記述から読み出し制御線および書き込み制御線の識別子を特定する。次のステップ102で、アクセスのために選択されるアドレスの記述からアドレス信号の識別子を特定する。次のステップ103で、アドレスにより選択されたメモリ部のデータの入出力に関する記述からデータ入出力信号の識別子を特定する。ステップ104では、機能シミュレーションの実行と同時進行でメモリブロックをアクセスするテストベクトルを検出し、これに対応させてシミュレーションにおける仮想時刻（シミュレーション時刻）と、その時刻の書き込み制御線の状態と、読み出し制御線の状態と、選択アドレスと、選択アドレスへの書き込みデータまたは選択アドレスからの読み出しデータとをこれら識別子に基づいて抽出し、選択アドレスと選択アドレスへの入出力データ（書き込み／読み出しデータ）についてはLSIにおけるアドレス線とデータ入出力線の実配置に対応した配列順の2進数表現に変換してからメモリブロックの内部状態として出力する。

【0023】図2を参照すると、故障解析支援装置12は、識別子特定手段21と、内部状態抽出手段22とを有している。識別子特定手段21は、機能記述ファイル10内のLSIのメモリブロックの機能記述を構文解析し、抽出対象の配線もしくは信号の識別子を特定する。すなわち、識別子特定手段21は、メモリブロックの機能記述からメモリ容量を抽出し、読み出し制御線および書き込み制御線の識別子を特定し、アドレス信号の識別子を特定して保存する。内部状態抽出手段22は、機能シミュレータ11でのシミュレーションの実行と同時進行でメモリブロックをアクセスするテストベクトルを検出し、検出されたテストベクトルに対応させてシミュレーションにおける仮想時刻と、その時刻の書き込み制御線の状態と、読み出し制御線の状態と、選択アドレスと、選択アドレスへの書き込みデータまたは選択アドレスからの読み出しデータとを識別子特定手段に保存された識別子情報に基づいて抽出し、選択アドレスと選択アドレ

スへの入出力データ（書き込み／読み出しデータ）についてはLSIにおけるアドレス線とデータ入出力線の実配置に対応した配列順の2進数表現に変換してから内部状態ファイル13に出力する。内部状態ファイル13に出力する代わりに表示装置に表示してもよく、またはプリンタで直接リスト出力してもよい。

【0024】なお、故障解析支援装置12は、専用のハードウェアとして構成してもよいが、汎用のコンピュータと、識別子特定手段21に相当する機能のプログラムと、内部状態抽出手段22に相当するプログラムとを用いても実現できる。

【0025】図3(a)は、このようにして作成された内部状態ファイル13をリスト出力した内部状態リストの一例である。クロックで計測したシミュレーション時刻（/SIMCLK）毎に、書き込み制御線（/WRITE）の状態、読み出し制御線（/READ）の状態、選択アドレス（/ADDR）、選択アドレスから入出力されるデータ（/DATA）のそれぞれの期待値が2進数表現で出力される。なお、“Z”はハイインピーダンス状態であることを示す。

【0026】図3(b)は、LSIの8ビットのデータバス配線DATA_BUS[0]～DATA_BUS[7]の信号電位実測結果の一例を示す図である。斜線を施した配線は高電位が実測された配線を示す。データバスの実測結果を高電位が“1”で低電位が“0”として2進数表現すると（10101010）となり、図3(a)の時刻110に対応する実測結果である場合には、データ（/DATA）と一致していると一目瞭然に容易に判断できる。また、図3(a)の時刻1210に対応する実測結果である場合には、データ（/DATA）とは一致していないことも容易に判断できる。

【0027】図4は、故障解析のフロー図である。作成された内部状態リストを用いて故障箇所の追跡を行うときには、まず、ステップ110で、メモリブロック外部の信号線で故障データが伝播した信号線を追跡対象として定め、この追跡信号線から遡って故障伝搬経路を追跡する。次のステップ111で、追跡された信号線がメモリブロックに到達するか否かを判断する。メモリブロックに到達しない場合にはステップ115に進み、メモリブロックに到達する場合には次のステップ112に進む。ステップ112では、メモリブロックの読み出し制御線の電位、書き込み制御線の電位、アドレス線の電位を電子ビームテスト等を用いて実測し、図3(a)に出力された内部状態リストで対応するシミュレーション時刻でのそれぞれの期待値と比較して一致するか否かを判断する。一致しなかった場合にはその信号線が故障伝搬経路にあるとしてステップ115に進む。一致した場合には次のステップ113に進む。

【0028】ステップ113では、図3(a)に出力された内部状態リストを参照してアドレス線で指定された

選択アドレスに保持されたデータが書き込みされる時刻にまで遡り、その時刻をメモリに故障が伝搬した時刻として特定する。この故障伝搬時刻に設定してメモリに入力されるデータ入力信号線（データバス）の電位を実測し、図3（a）の内部状態リストの同時刻における入力データ（/DATA）の期待値と比較して一致するか否かを判断する。一致しなかった場合にはデータ入力信号線が故障伝搬経路にあるとしてステップ115に進む。一致した場合には次のステップ114に進む。次のステップ114では、ステップ113で特定した故障伝搬時刻におけるメモリブロックの制御線、すなわち、読み出し制御線、書き込み制御線およびアドレス線の電位を実測し、図3（a）に出力された内部状態リストで対応するシミュレーション時刻でのそれぞれの期待値と比較して一致しない信号線を検出し故障伝搬経路として特定し、ステップ115に進む。

【0029】ステップ115では、故障伝搬経路のさらなる遡行が可能か否かを判断する。まだ故障伝搬経路を溯れるときにはステップ110に戻って故障伝搬経路の追跡を再度繰り返す。故障伝搬経路を溯れないときには故障箇所を特定できたとして故障解析を終了する。

【0030】次に、8ビット×2048ワードのメモリブロックをLSI設計言語の1つであるVerilog-HDL（Verilogはケーデンス・デザイン・システムズ社の登録商標である）で機能記述した図5と、機能シミュレーション実行時にメモリブロックの内部状態を抽出して内部状態ファイルに格納する制御を記述した図6とを参照して図1のフローを具体的に説明する。

【0031】まず、ステップ100では、機能記述された機能ブロックの構造を解析し、レジスタ宣言文の中で、`reg [7:0] iram8 [0:11'h7fff]`と定義された記述（D100）からメモリレジスタ（メモリブロック）が8bit×2048wordの容量であることを抽出し、記述レンジ指定記述に挟まれた文字列iram8をメモリレジスタに割当てられた識別子として抽出する。

【0032】次のステップ101では、メモリレジスタiram8の特定のアドレスへデータを読み出す記述（D101）とデータを特定のアドレスへ書き込む記述（D101）とから、読み出し制御信号名rdと書き込み制御信号名wrとを抽出する。

【0033】次のステップ102では、メモリブロックのアドレスを配列変数で指定する記述（D102）で、メモリレジスタiram8の選択アドレスを指定する`addr_in [10:0]`にアドレスデータを渡しているADDRをアドレス信号線の信号名と抽出する。

【0034】次のステップ103では、メモリレジスタiram8のデータ入出力に関する記述（D103）で、選択アドレスからデータを読み出すレジスタと、選択アドレスにデータを書き込むレジスタとがデータを受

け渡しているDATA_BUSをデータ信号線名として抽出する。

【0035】次のステップ104における処理の具体例については、図6の出力制御記述にしたがって説明する。ステップ104では、まず、設計情報の最上位構造から書き下して読み出し制御線と書き込み制御線とアドレス線とデータ線を指定（D1040）する。このときに、データ入出力線とアドレス線ではLSIにおけるバス配線の配置に対応した配列に置換しておく。図6の記述は、アドレスのMSBからLSBの順でアドレス線が配置され、続いてデータのMSBからLSBの順でデータ入出力線が配置されている例である。次に、データを格納する内部状態ファイル名（`dump_file`）を指定してファイルを開き（D1041）、内部状態ファイルに格納する状態情報としてシミュレーション実行時の仮想的な時間経過を計測する変数（`clk_count`）を定義し（D1042）、出力ファイルの先頭行に、最初にシミュレーション時刻を出力し続いて各信号線の名称を予め定めた順序で出力する（D1043）。続いて、シミュレーションを実行し、シミュレーションが進行する毎にシミュレーション時刻と識別子に基づいて抽出した各信号線の状態情報とを内部状態ファイルに所定の順序で格納する（D1044）。

【0036】内部状態リストを参照することにより、任意のシミュレーション時刻におけるアドレス線ADDR[10:0]の保持する2進数表現の期待値を容易に特定することができ、同様に、任意のシミュレーション時刻においてデータ入出力線DATA_BUS[7:0]の保持する2進数表現の期待値を容易に特定することができる。シミュレーション時刻に対応する時刻の不良品における選択アドレスはアドレス線の電位を実測することによって得られる。同様に、シミュレーション時刻に対応する時刻の不良品における書き込み/読み出しデータはデータ入出力線の電位を実測することによって得られる。

【0037】LSIのレイアウトでは、通常、メモリブロックのアドレス線は、アドレスのLSBからMSBに向けて、またはMSBからLSBに向けて、順番に隣接して並行に配置される。同様に、メモリブロックのデータ入出力線は、データのLSBからMSBに向けて、またはMSBからLSBに向けて、順番に隣接して並行に配置される。したがって、通常、簡単な操作により、アドレス線およびデータ入出力線の実際の配置順序と内部状態ファイルに格納するアドレス線およびデータ入出力線の2進数表現情報の格納順序を一致させることが可能である。

【0038】本実施形態においては、機能記述のみでゲート記述レベルの情報を持たないメモリブロックが故障信号の伝播経路上に存在する場合においても故障伝搬経路が制御信号であるかまたはデータ入出力であるかを分

離して追跡することが可能となる。また、内部状態リストにLSIにおけるアドレス線とデータ入力線の実際の配置順序と一致するようにアドレス線の期待値とデータ入出力線の期待値との配列順序を変更して表示することにより、図3に例示したように実測値と期待値との比較が飛躍的に容易となり、故障伝搬経路の追跡において、期待値ファイルを用いて配線毎に個別に信号線が保持する値を検証する場合に比較して追跡作業の労力と時間の大幅な低減が可能となる。

【0039】次に本発明の第2の実施形態について説明する。第2の実施形態では、機能記述されたメモリブロックについて、LSIチップの機能テストにおいてアクセスしたアドレス部分のみを、トランジスタ等の素子と、素子間の接続とで表現されたトランジスタレベルのネットリストを作成する。元の機能記述のメモリブロックを作成したトランジスタレベルのネットリストに置換することにより、市販の故障解析ツール等の公知の故障解析技術を用いて故障追跡することが可能となる。

【0040】図7は、機能テストでアクセスしたアドレスに関係した部分のみのトランジスタレベルのネットリストを作成フロー図である。

【0041】図7では、まず、ステップ200で、機能テストにおいて故障が出力されるテストベクトルまでにアクセスしたメモリアドレスをテストベクトル毎に取得する。次のステップ201で、LSIチップのレイアウトデータのデータ入出力線からレイアウトデータのトレースを開始し、電源線または接地線に到達したときには一旦停止して別の経路をトレースし、ワード線数以上の分岐を持つ配線に到達した時点でトレースを停止する。読み出し回路への入力線のようにワード線数以上の分岐を持つ配線に到達した場合には、この配線を分岐配線とし、データ入出力線から分岐配線までの経路のレイアウトデータを抽出して保存する。

【0042】次のステップ202では、ステップ200において抽出されたアドレスのうち最後にアクセスされたアドレスを選択する。最初の選択では、故障が出力されたテストベクトルに時間的に最も近いメモリアccessのアドレスが選択されることになる。次のステップ203では、ステップ202において選択されたアドレスから選択されるワード線を特定し、特定されたワード線からトレースを開始し、電源線または接地線に到達したときは一旦停止して別の経路をトレースする。メモリブロック内をステップ201において保存された分岐配線に到達するまでトレースを継続し、経路に存在するレイアウトデータを抽出する。

【0043】次のステップ204では、ステップ202で抽出されたレイアウトデータとステップ201で保存されたレイアウトデータからトランジスタレベルのネットリストを抽出し、アクセスしたメモリセルに関する部分回路のネットリストを作成する。部分回路には、ア

ドレス入力回路、選択ワード線、選択メモリセル、書き込み制御線、読み出し制御線データ入出力線、書き込み回路および読み出し回路を含む。次のステップ205で、ステップ200において抽出したアクセスについてネットリストの作成が完了したかを判断し、作成未了のアクセスが残っている場合にはステップ202に戻る。全てのアクセスに対応してネットリストの作成が完了したと判断された場合にはステップ206に進む。ステップ206では、抽出された全てのネットリストを合成し、複数のネットリストで重複している部分は1個に整理して置換用のトランジスタレベルネットリストを作成して終了する。

【0044】図8は、機能テストでアクセスしたアドレスに関係した部分のみのトランジスタレベルのネットリストを作成する故障解析支援装置のブロック図である。故障解析支援装置31は、アクセスアドレス保持手段41と、経路データ抽出手段42と、ネットリスト作成手段43とを備えている。アクセスアドレス保持手段41は、機能テスト結果30から故障が出力されるテストベクトルまでのメモリブロックへのアクセスを抽出してアクセス対象のメモリアドレスとともに保持する。経路データ抽出手段42は、アドレス入力から選択されるワード線を特定し、特定された選択ワード線からワード線の個数以上の分岐を持つ分岐配線に到達するまでメモリブロック内をトレースして経路に存在するレイアウトデータを抽出する。ネットリスト作成手段43は、抽出されたレイアウトデータからトランジスタレベルのネットリストを抽出しアクセスしたアドレスに対応するネットリストを作成するとともに複数のネットリストを1個のネットリストに合成する。図8の解析支援装置31を用いて図7のネットリストを作成フローにしたがって手順を実行することにより、LSIチップの機能テストにおいてアクセスされたアドレスの部分のみをトランジスタレベルで記述したネットリストを作成することができる。

【0045】なお、故障解析支援装置31は、専用のハードウェアとして構成してもよいが、汎用のコンピュータと、アクセスアドレス保持手段41に相当する機能のプログラムと、経路データ抽出手段42に相当するプログラムと、ネットリスト作成手段43に相当するプログラムとを用いても実現できる。

【0046】次に、図9および図10を用いて図7のフロー図に沿って具体的に説明する。図9は、図5のVerilog-HDLで機能記述されたメモリブロックiram8のアドレスデコーダ部の回路図である。11本のアドレス信号ADDR[0]~ADDR[10]により2048本のワード線M0000~M2047のうち1本を選択する。

【0047】図10は、同様に図5で機能記述されたメモリブロックiram8のメモリセルをラッチで表現した回路の一部で、ビット[0]の部分を示した回路図で

ある。メモリブロック `iram8` は、ラッチ `A2048` とラッチ `A4096` とで1個のメモリセルが構成され、同様にラッチ `A4095` とラッチ `A6143` とで構成されるメモリセルまで計2048個のメモリセルを有する。書き込み時にはAND回路 `A0000` ~ `A2048` の内の選択されたワード線が入力するAND回路と接続するメモリセルに `data_in[0]` を書き込む。読み出し時には、2048個の3ステートのバッファ回路 `A6144` ~ `A8191` の内の選択されたワード線により活性化されたバッファ回路に接続するメモリセルから `data_out[0]` を読み出す。なお、図9および図10において、CLKはクロック信号、WEは入力制御信号、OEは出力制御信号、CSはメモリブロック制御信号である。

【0048】以下、図7に沿って図9と図10のネット名を用いて説明する。ステップ200で故障出力までにアクセスした全てのアドレスを抽出した後に、ステップ201で、図10に対応するLSIチップのレイアウトデータを `DATA_BUS[7:0]` からトレースを開始し、ワード数の分岐を持つ `data_in[0]` および `M165` に到達した時点でネットリストの抽出を一時停止する。ステップ203では、図10の `ADDR[0]` ~ `ADDR[10]` の入力の“0”または“1”に応じてレイアウトデータを抽出する経路を選択する。例えば、アドレス入力がすべて“0”ならば、2048個の11入力NAND回路の内のワード線 `M0000` に接続するNAND回路のみが“1”を出力する。この場合には、ワード線 `M0000` に接続しているメモリセル (`A2048` および `A4096`) のレイアウトデータを抽出し、さらにトレースを継続してこのメモリセルに接続する制御線とデータ入出力線のレイアウトデータを抽出する。トレースがワード数の分岐を持つ `data_in[0]` および `M165` に到達した時点でトレースを完了する。

【0049】ステップ206で、機能テストで故障出力までにアクセスされた全てのアドレスに対して抽出されたネットリストを合成して置換用ネットリストを作成した後に、(図示していないが) さらに次のステップとしてこの置換用ネットリストと機能記述されたメモリブロックを置換した設計情報を用いて機能検証シミュレーションを実行し、機能記述を用いたシミュレーションと同じ期待値が得られるかどうかを検証しておくことが、後の故障解析の正当性を裏付けるために望ましい。

【0050】図11は、図7のフローにしたがって作成したメモリブロックのトランジスタレベルネットリストを用いての故障解析フロー図である。まず、ステップ210で、メモリブロックの機能記述を、作成したメモリブロックのトランジスタレベルネットリストに置換して故障診断ソフトを実行する。故障診断ソフトとしては、故障辞書法を用いたもの、バックトレース法を用いたもの

の何れをも使用可能である。

【0051】次のステップ211で、故障診断により故障候補箇所が少数に絞り込まれたかを判断する。故障診断で故障候補箇所を少数に絞り込んだときには、ステップ212に進み、故障候補箇所での故障信号が発生するように設計情報を意図的に変更して機能シミュレーションを実行する。次のステップ213で、シミュレーション出力が実際の故障信号出力と一致するか否かを判断し、一致する場合には故障箇所及び故障原因を特定できたとして故障解析を終了する。ステップ213で一致しなかった場合にはステップ212に戻り、別の故障候補箇所についてステップ212を実行する。

【0052】ステップ211で、故障候補箇所が多数存在し、故障を埋め込んでのシミュレーションに多大な時間を要するので実質的に不可能な場合には、故障診断ソフトの性能限界と判断して故障候補箇所の絞り込みを放棄し、シミュレーションの期待値と実測値とを照合しながら故障伝搬経路を追跡する物理的な故障解析を実行して故障箇所を特定する。

【0053】本実施形態においては、メモリセルブロックの記述を機能記述からトランジスタレベルのネットリストに置換することができるので、市販品を含めた公知の故障診断ソフトを使用して容易且つ効率的に故障解析を進めることが可能となる。また、機能テスト時にアクセスしたアドレスに関係する部分のレイアウトデータのみを抽出してからトランジスタレベルのネットリストを作成するので、メモリブロックの全レイアウトデータからメモリブロックの完全なトランジスタ記述ネットリストを抽出作成しシミュレーションに用いる従来の技術と比較して、ネットリスト作成に要する時間をおよそ(アクセスアドレス数/全アドレス数)に短縮でき、シミュレーション時間も短縮できる。特にメモリ容量の大きいメモリブロックに適用した場合には、これらの短縮効果が顕著となる。このように、第2実施形態においても、機能記述されたメモリブロックの故障追跡作業における労力と時間の大幅な低減が可能である。

【0054】

【発明の効果】以上に説明したように、本発明を適用することにより、機能記述のみでゲート記述レベルの情報を持たないメモリブロックが故障信号の伝播経路上に存在する場合においても、メモリブロック内部のアクセスされたアドレスのメモリセルにまで立ち入った故障伝播経路の追跡を実用的な労力および時間で実行することが可能となる。

【図面の簡単な説明】

【図1】本発明の第1実施形態におけるメモリブロックの内部状態ファイルの作成フロー図である。

【図2】内部状態ファイルを作成する故障解析支援装置を含むシステムのブロック図である。

【図3】(a)は、内部状態リストの一例であり、

(b) は、信号電位実測結果の一例を示す図である。

【図4】故障解析のフロー図である。

【図5】8ビット×2048ワードのメモリブロックの機能記述の一例である。

【図6】機能シミュレーション実行時にメモリブロックの内部状態を抽出して内部状態ファイルに格納する制御記述の一例である。

【図7】本発明の第2実施形態のトランジスタレベルのネットリスト作成フロー図である。

【図8】トランジスタレベルのネットリストを作成する故障解析支援装置のブロック図である。

【図9】メモリブロックのアドレスデコーダ部の回路図

である。

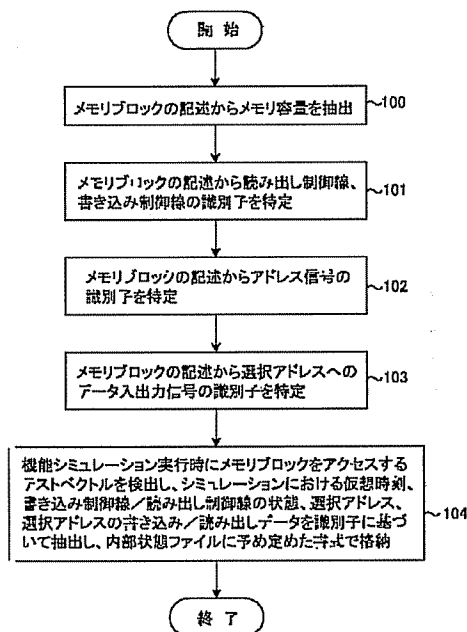
【図10】メモリブロックのメモリセルをラッチで表現した回路の一部分を示す回路図である。

【図11】トランジスタレベルネットリストを用いての故障解析フロー図である。

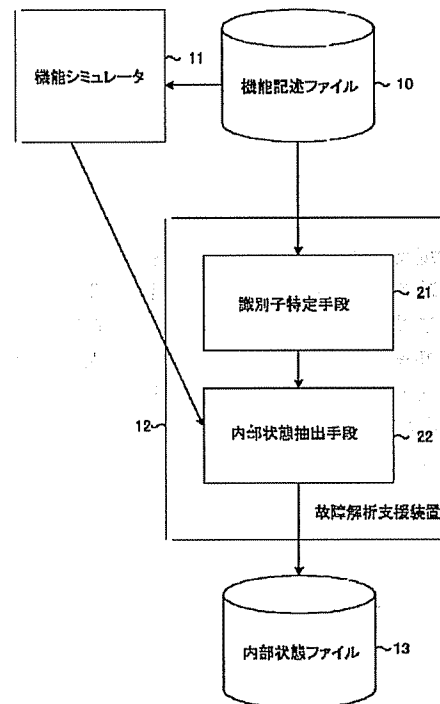
【符号の説明】

- 12, 31 故障解析支援装置
- 21 識別子特定手段
- 22 内部状態抽出手段
- 41 アクセスアドレス保持手段
- 42 経路データ抽出手段
- 43 ネットリスト作成手段

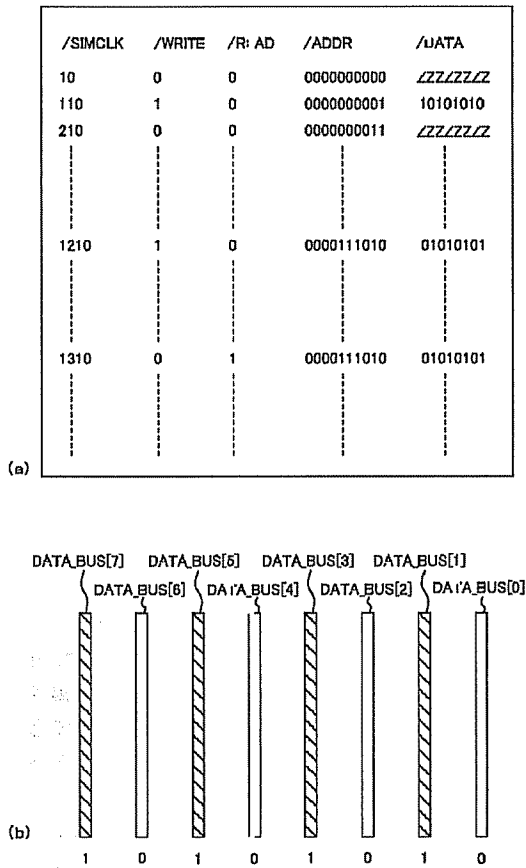
【図1】



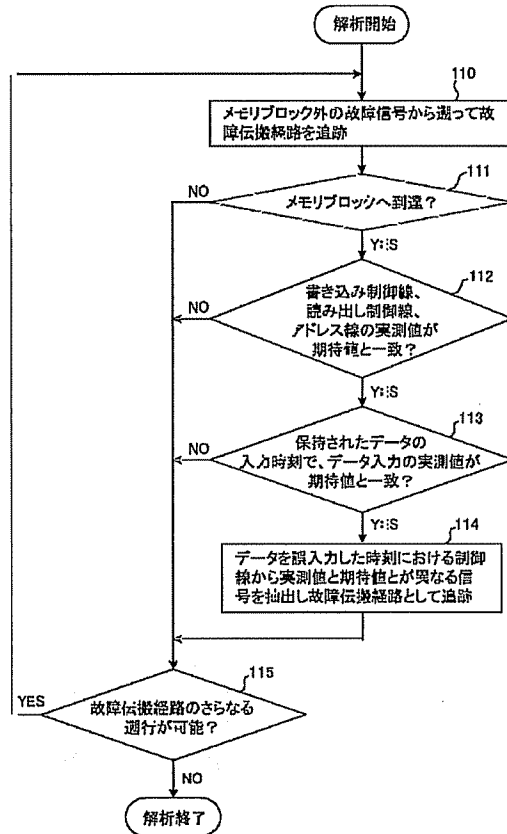
【図2】



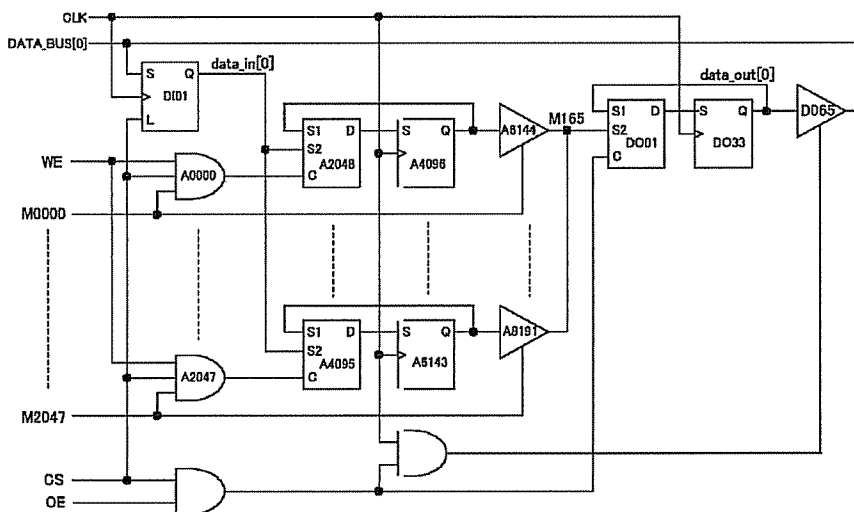
【図3】



【図4】



【図10】



【図5】

```

module IRAM( DATA_BUS, ADDR, CS, WE, OE, CLK);
inout [7:0] DATA_BUS;
input [10:0] ADDR;
input  CS, WE, OE, CLK; // chip selector, write enable, read enable

wire [10:0] addr_in;
wire rd, wr;
reg [7:0] iram8[0:11'h7f]; // 2KB } D100
reg [11:0] addr;
reg [7:0] data_in, data_out;

assign rd = CS & OE & CLK;
assign wr = CS & WE & CLK;

always @(ADDR or (CS & CLK))
if (CS & CLK)
addr_in = ADDR; } D102

always @(d or (CS & CLK)) // data latch
if (CS & CLK)
data_in[7:0] = DATA_BUS; } D103

assign DATA_BUS = (rd == 1'b1) ? data_out : 8'bz;

always @(posedge CLK) // read/write control
if (rd) begin // read
data_out[7:0] = iram8[addr_in[10:0]];
end
else (wr) begin // write
iram8[addr_in[10:0]] = data_in[7:0];
end
endmodule

```

【図6】

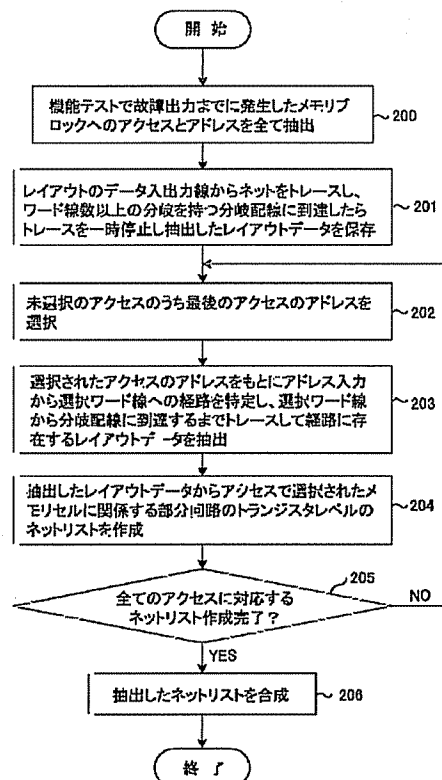
```

module ADDR_DATA_DUMP0
wire address[10:0] = 'upper_memory_block.ADDR;
wire data[7:0] = 'upper_memory_block.DATA_BUS;
wire wr = 'upper_memory_block.wr;
wire rd = 'upper_memory_block.rd; } D1040
integer spal;
integer clk_count;
'define dump_file "[directory]/memory.dump"
initial
begin } D1041
spal = $fopen("dump_file");
initial
begin
clk_count=0;
forever #1 clk_count = clk_count + 1; } D1042
end
$display(spall, "IPAT /WRITE /READ /ADDRESS /DATA"); } D1043
forever begin
@('SIMCLK);
#10;
$display(spall, "%5.0f %b %b %b %b", clk_count, wr, rd, address, data); } D1044
end //end-forever

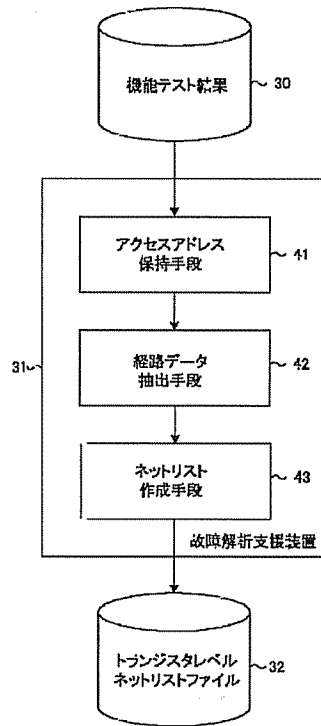
end // end init
endmodule

```

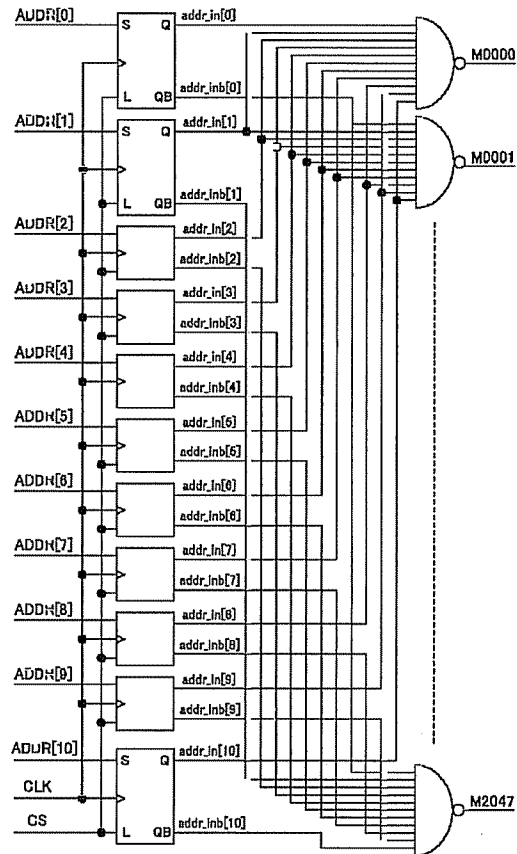
【図7】



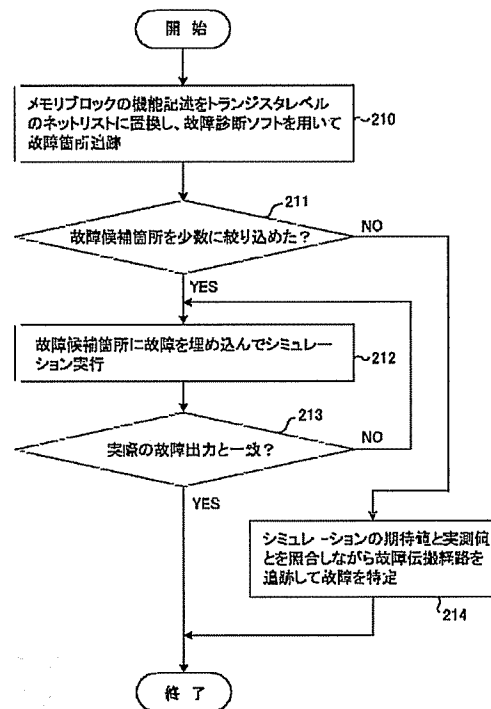
【図8】



【図9】



【 図 11 】



フロントページの続き

(51)Int. Cl.⁷

識別記号

F I
G O I R 31/28(参考)
F